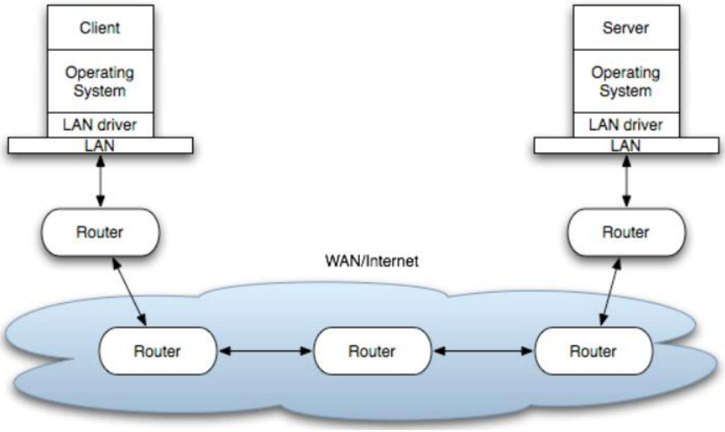
Configure and Manage Azure Virtual Networks

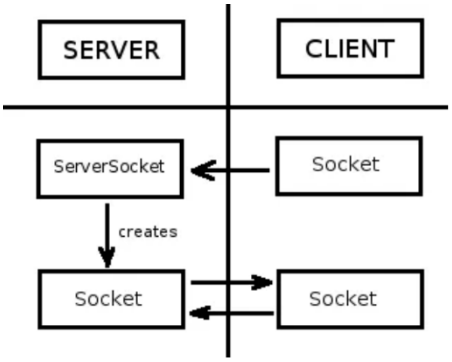
- Overview of Azure Networking
- Virtual Network Benefits
- Understanding Network Resources
- Implement and manage virtual networking
    - Create a VNet using Azure Portal
    - Create a Subnet
    - Configure private and public IP addresses
    - Create Network Interface Card with public, and private IP addresses
    - Create a Virtual Machine
- Setup Network Security Group
    - Create security rules
    - Associate NSG to a subnet or network interface
    - Identify required ports
    - Evaluate effective security rules
    - Application Security Groups
- Understanding Azure DNS
    - Configure Azure DNS
    - DNS Zones
    - DNS Records and Record Sets
    - DNS Resolution
- Network Routing Table
    - System Routes
    - User Defined Routes
    - Creating User Defined Route Table
    - Create and Associate Route
- Create connectivity between virtual networks
    - Overview
    - Create a Point to Site VPN
    - Create and configure VNET to VNET
    - Verify virtual network connectivity
    - Create and Configure VNET peering

OSI Layers in Network Communication

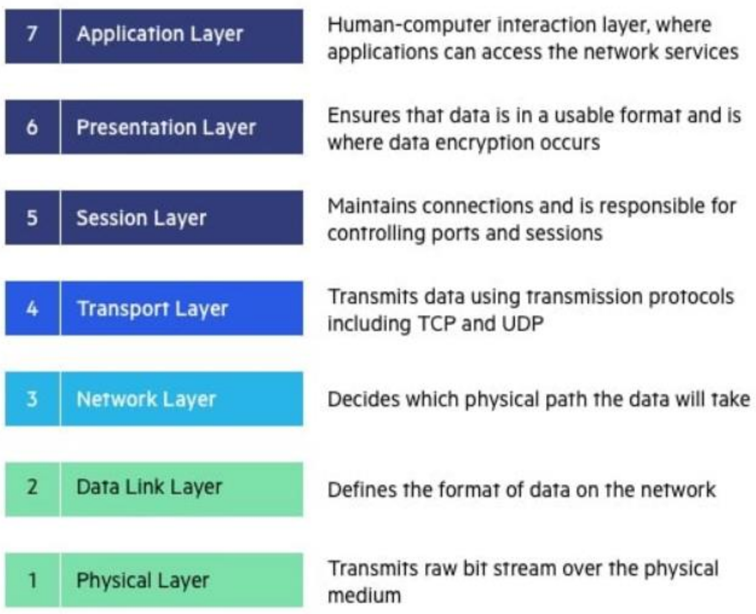


Socket = IP + Port No

HTTP=80, HTTPS=443, FTP=21, SQL Server DB = 1433, RDP=3389, SSH=22…



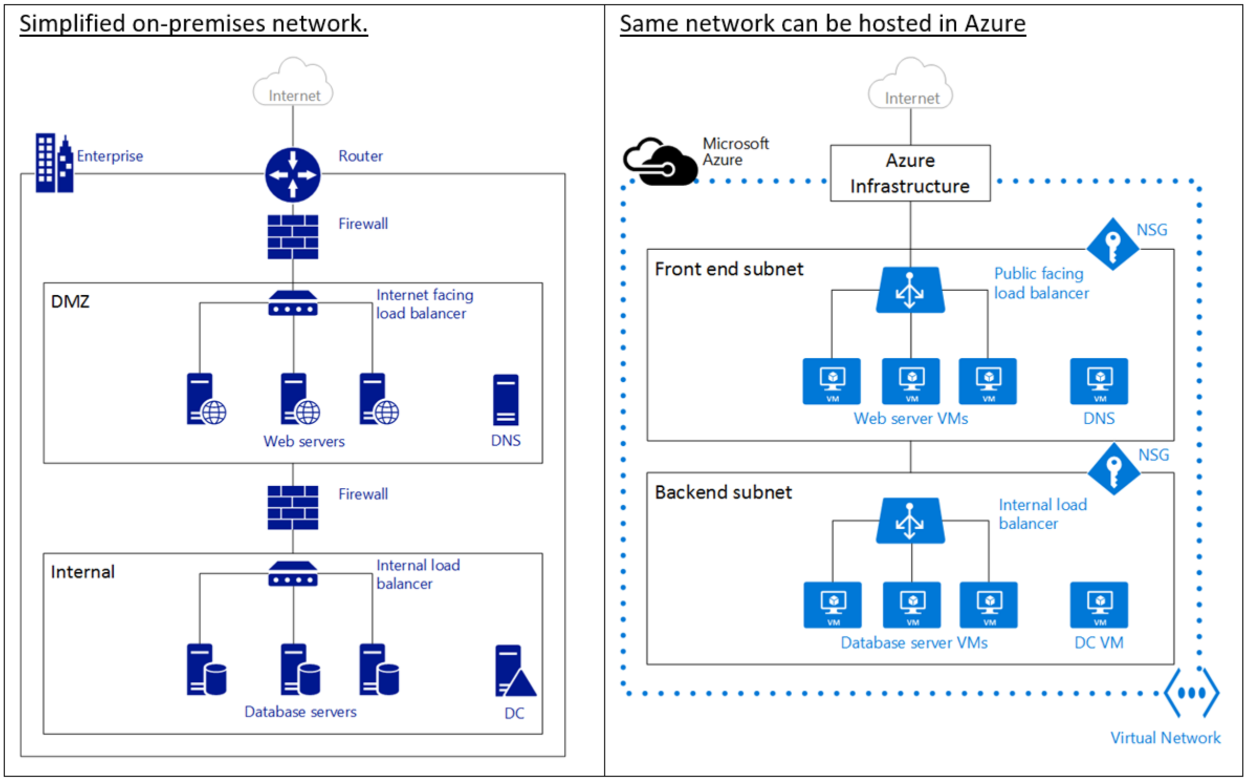
OSI Layers:



7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium

Overview of Azure Networking

- An Azure virtual network (VNet) is a representation of your own network in the cloud.
- It is **a logical isolation** of the Azure cloud dedicated to your subscription. You can fully control the IP address blocks, DNS settings, security policies, and route tables within this network.
- You can also further segment your VNet into **subnets** and launch Azure virtual machines (VMs).
- You can connect the virtual network to your on-premises network using one of the connectivity options available in Azure. In essence, you can expand your network to Azure, with complete control on IP address blocks with the benefit of enterprise scale Azure provides.



*In computer **networks**, a **DMZ** (**demilitarized zone**) is a physical or logical **sub-network** that separates an internal local area **network** (LAN) from other untrusted **networks**, usually the Internet.

Notice how the Azure infrastructure takes on the role of the router, allowing access from your VNet to the public Internet without the need of any configuration. Firewalls can be substituted by Network Security Groups (NSGs) applied to each individual subnet. And physical load balancers are substituted by internet facing and internal load balancers in Azure.

Azure VNet Pricing:

- There is **no extra cost** for using Virtual Networks in Azure.
- The compute instances launched within the Vnet will be charged the standard rates as described in Azure VM Pricing.
- The VPN Gateways and Public IP Addresses used in the VNet will also be charged standard rates.

Virtual Network Characteristics

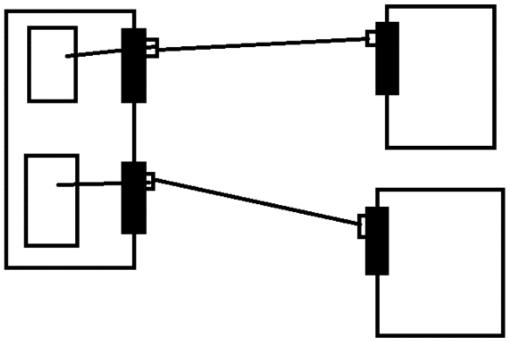
- **Isolation**. VNets are completely isolated from one another. That allows you to create disjoint networks for **development, testing, and production** that use the same CIDR address blocks.
- **Connectivity**. VNets can be connected to each other, and even to your on-premises datacenter, by using a site-to-site VPN connection, or ExpressRoute connection.
- **Access to the public Internet**. All VMs in a VNet can access the public Internet by default. You can control access by using Network Security Groups (NSGs).
- **Security**. Traffic entering and exiting the virtual machines in a VNet can be controlled using Network Security groups and Azure Firewall.
- **Access to VMs within the VNet**. VMs can be launched in the same virtual network and they can connect to each other using private IP addresses even if they are in different subnets without the need to configure a gateway or use public IP addresses.
- **Name resolution**. Azure provides internal name resolution for IaaS VMs deployed in your VNet. You can also deploy your own DNS servers and configure the VNet to use them.

Note: The most important thing about Azure virtual networks is that you cannot add an existing virtual machine to a newly created virtual network. It is important that if you want to leverage virtual networking in Azure that you must create the virtual networks **BEFORE** creating your virtual machines! Don’t miss this important step. You’ll be disappointed if you’ve spent a lot of time setting up a virtual machine and later find that you can’t move it to a virtual network.

- **Subnets**: Subnet is a **range of IP addresses** in the VNet, you can divide a VNet into multiple subnets for organization and security. VMs deployed to subnets (same or different) within a VNet can communicate with each other without any extra configuration. You can also configure **route tables and NSGs** to a subnet.

- **Network Interface Card (NIC):** VMs communicate with other VMs and other resources on the network by using virtual network interface card (NIC). Virtual NICs configure VMs with private and optional public IP address. VMs can have more than one NIC for different network configurations.

  Note: VMs can have more than one NIC adapter that links the VM with the virtual network. The number of NICs you can attach to a VM depends on its size. For example, a VM that is based on a D2 size can have 2 NICs, and a D4-based VM can have a maximum of 16 NICs. Multiple NICs configuration is common for virtual appliances that provide additional control of traffic in virtual networks.



- **Network Security Group** (NSG): You can create NSGs to control **inbound and outbound** access to network interfaces (NICs), VMs, and subnets. Each NSG contains one or more rules specifying whether or not traffic is **allowed or denied** based on **protocol, source IP address, source port, destination IP address, and destination port.**

Understanding IP Address Space

- **IP addresses**: There are two types of IP addresses assigned to resources in Azure: *public* and *private*.
  a. **Public IP Addresses** allow Azure resources to communicate with Internet and other Azure public-facing services like Azure Redis Cache.
  b. **Private IP Addresses** allows communication between resources in a virtual network, along with those connected through a VPN, without using an Internet-routable IP addresses.

Preferred IP Series for Intranets (Private IP):

Small Network1: 192.168.0.X – for 2^8 Systems – IP Address Range = 192.168.0.0/24 (Only last byte changes)

Small Network2: 192.168.1.X –for 2^8 Systems – IP Address Range = 192.168.1.0/24 (Only last byte changes)

Large Network: 172.16.X.X – for 2^{16} Systems - IP Address Range = 172.16.0.0/16 (last 2 bytes change)

Very Large Network: 10.X.X.X – for 2^{24} Systems – IP Address Range = 10.0.0.0/8 (last 3 bytes change)

Classless Inter-Domain Routing (**CIDR) notation** is a compact representation of an IP address and its associated routing prefix. The **notation** is constructed from an IP address, a slash ('/') character, and a decimal number. The number is the count of leading 1 bits in the routing mask, traditionally called the network mask.

202.123.56.123 to 202.123.56.123 = 202.123.56.123/32

0.0.0.0 to 255.255.255.255 = 0.0.0.0/0

```
192.168.0.0 to 192.168.0.255 = 192.168.0.0/24 = 256

192.168.170.0 to 192.168.170.255 = 192.168.170.0/24 = 256

172.16.0.0 to 172.16.255.255 = 172.16.0.0/16 = 256 * 256

202.34.0.0/16 = 202.34.0.0 to 202.34.255.255 = 256 * 256


10.0.0.0/16 = 10.0.0.0 to 10.255.255.255 = 256*256*256
  10.0.0.0/28
          10.0.0.4
          10.0.0.5
          10.0.0.6
          10.0.1.5
  10.0.1.0/24
          10.0.1.4
          10.0.1.5
          10.0.1.6
  10.0.2.0/24
  10.0.3.0/24
  10.0.4.0/24
10.1.0.0/16
  10.1.0.0/24
          10.1.0.4
          10.1.0.5
          10.1.0.6
          10.1.1.5
  10.1.1.0/24
          10.1.1.4
          10.1.1.5
          10.1.1.6
  10.1.2.0/24
  10.1.3.0/24
  10.1.4.0/24
10.2.0.0/16
```

Public IP Addresses

- There are two methods in which an IP address is allocated to a *public* IP resource - ***dynamic*** or ***static.***
    - In the **dynamic** allocation method the IP address is **not** allocated at the time of its creation. Instead, the public IP address is allocated when you start (or create) the associated resource (like a VM or load balancer). The IP address is released when you stop (or delete) the resource. This means the IP address can change.
    - In the **static** allocation method the IP address for the associated resource does not change. In this case an IP address is assigned immediately. It is released only when you delete the resource or change its allocation method to *dynamic*.
- Public IP addresses allow Azure resources to communicate with Internet and Azure public-facing services such as Azure Redis Cache, Azure Event Hubs, SQL databases and Azure storage.
- In Azure Resource Manager, a public IP address is a resource that has its own properties. You can associate a public IP address resource with any of the following resources:
    - Internet-facing Virtual machines (VM)
    - Internet-facing load balancers
    - VPN gateways
    - Application gateways
- Public IP address is paid service.

Private IP Addresses

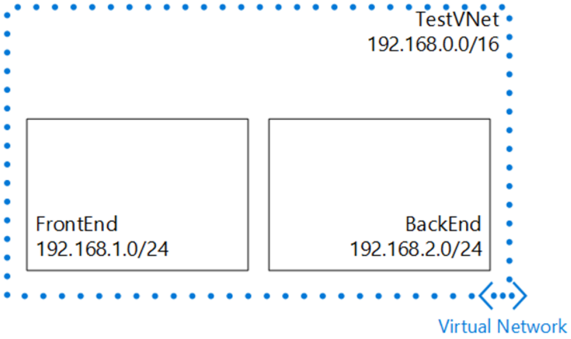
1. IP address is allocated from the address range of the subnet to which the resource is attached.
2. The default allocation method is dynamic, where the IP address is automatically allocated from the resource's subnet (using DHCP). This IP address can change when you stop and start the resource.
3. You can set the allocation method to static to ensure the IP address remains the same. In this case, you also need to provide a valid IP address that is part of the resource's subnet.
4. Private IP addresses allow Azure resources to communicate with other resources in a virtual network or an on-premises network through a VPN gateway or ExpressRoute circuit, without using an Internet-reachable IP address.
5. In the Azure Resource Manager deployment model, a private IP address is associated to the following types of Azure resources:
    - VMs
    - Internal load balancers (ILBs)
    - Application gateways

Create a Virtual Network (VNet) using the Azure portal

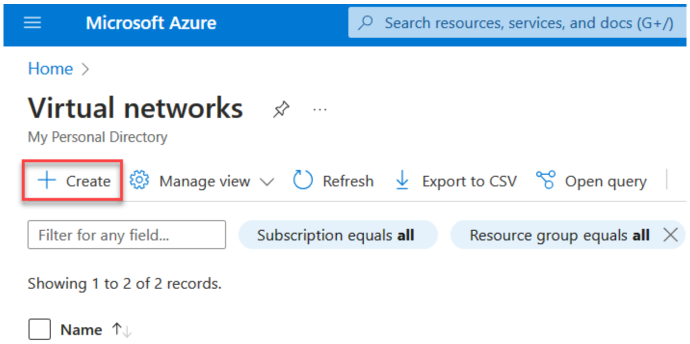
In this scenario we will create a VNet named **TestVNet** with a reserved CIDR block of **192.168.0.0/16**.

Your VNet will contain the following **subnets**:

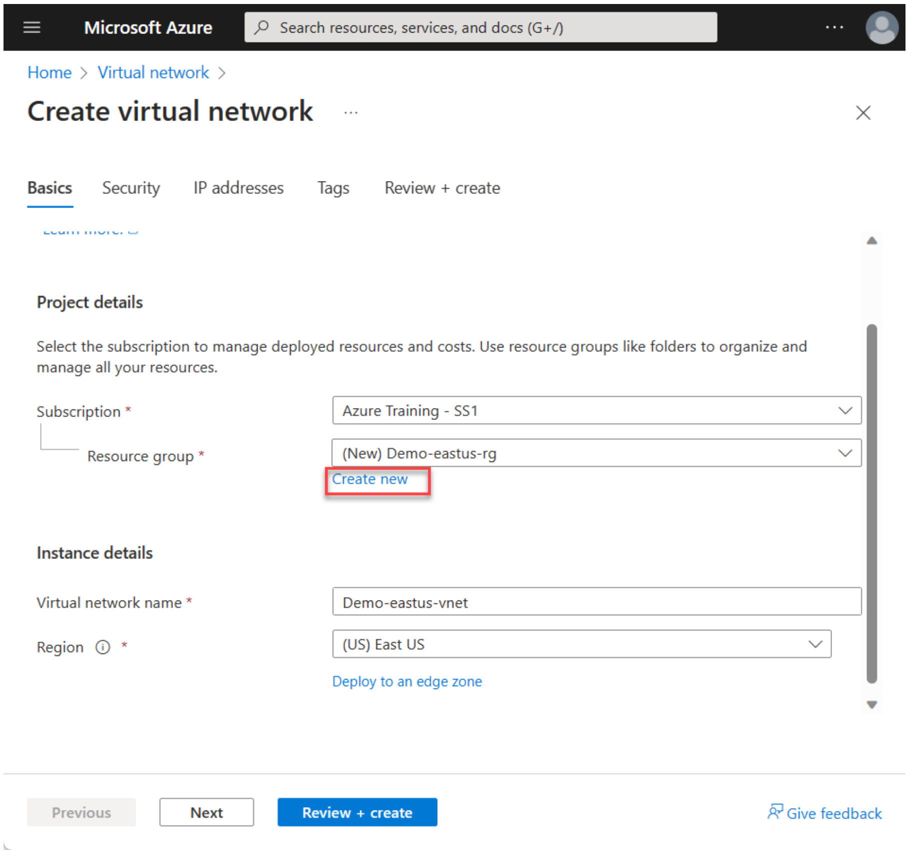
- **FrontEnd**, using **192.168.1.0/24** as its CIDR block.
- **BackEnd**, using **192.168.2.0/24** as its CIDR block.



1. Click Search → Virtual networks → **+Create**



2. Select your Subscription, Resource Group → Create new (Demo-eastus-RG), Virtual network name=Demo-eastus-vnet, Region = East US



Create virtual network

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure Training - SS1

Resource group * (New) Demo-eastus-rg

Create new

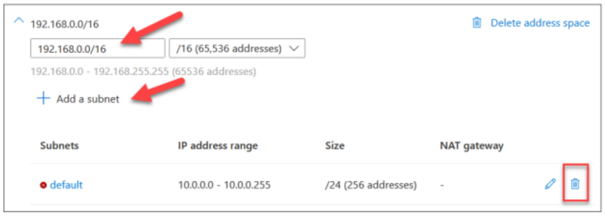
Instance details

Virtual network name * Demo-eastus-vnet

Region * (US) East US

Deploy to an edge zone

3. Go to IP address tab → Change CIDR block to 192.168.0.0/16, delete existing Subnet and click + Add a subnet



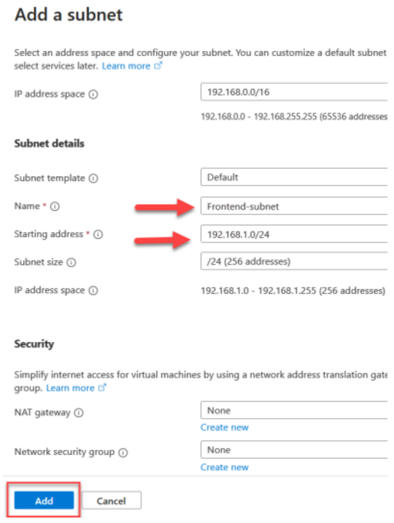
192.168.0.0/16

192.168.0.0/16 /16 (65,536 addresses)

192.168.0.0 - 192.168.255.255 (65536 addresses)

Subnets	IP address range	Size	NAT gateway
default	10.0.0.0 - 10.0.0.255	/24 (256 addresses)	-

4. IP Address Space=192.168.0.0/16, Subnet name="**Frontend-subnet**", Subnet Address Range=192.168.1.0/24, click Add button



Add a subnet

Select an address space and configure your subnet. You can customize a default subnet select services later. Learn more

IP address space: 192.168.0.0/16

192.168.0.0 - 192.168.255.255 (65536 addresses

Subnet details

Subnet template: Default

Name *: Frontend-subnet

Starting address *: 192.168.1.0/24

Subnet size: /24 (256 addresses)

IP address space: 192.168.1.0 - 192.168.1.255 (256 addresses)

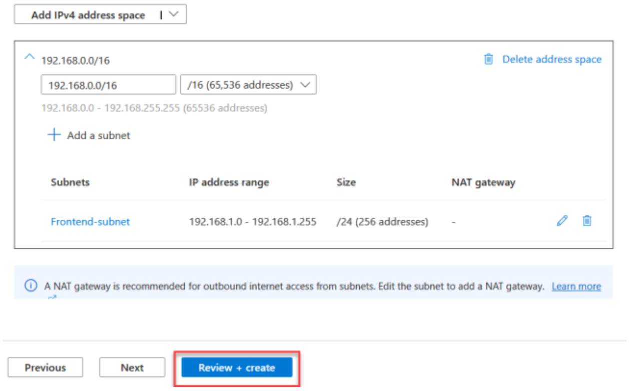
Security

Simplify internet access for virtual machines by using a network address translation gate group. Learn more

NAT gateway: None

Network security group: None

5. Click Review + create button, then Create button, wait for the VNet to be created. Click Go to resource button



192.168.0.0/16

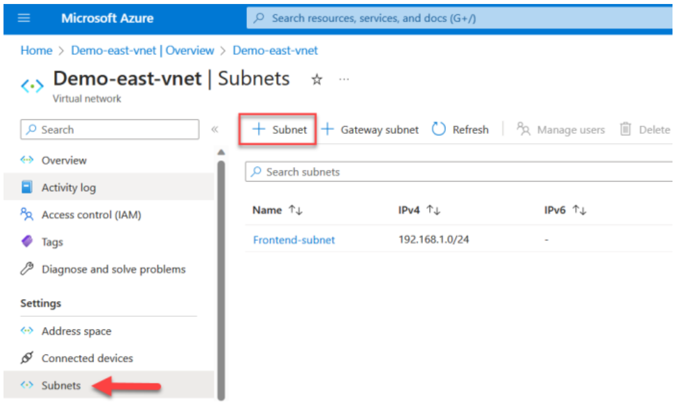
192.168.0.0/16 | /16 (65,536 addresses)

192.168.0.0 - 192.168.255.255 (65536 addresses)

Subnets	IP address range	Size	NAT gateway
Frontend-subnet	192.168.1.0 - 192.168.1.255	/24 (256 addresses)	-

A NAT gateway is recommended for outbound internet access from subnets. Edit the subnet to add a NAT gateway. Learn more

6. In the **Virtual network** blade, Settings section, click **Subnets** → + Subnet



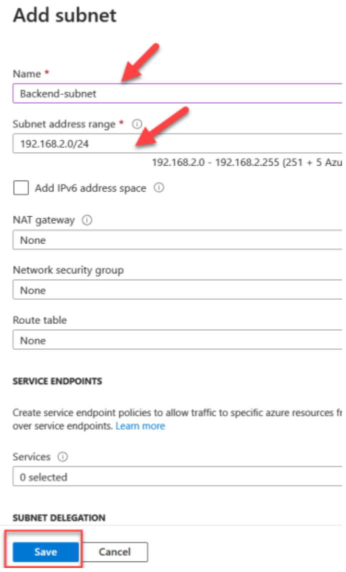
Home > Demo-east-vnet | Overview > Demo-east-vnet

Demo-east-vnet | Subnets

Virtual network

Name ↑↓	IPv4 ↑↓	IPv6 ↑↓
Frontend-subnet	192.168.1.0/24	-

7. Name=**Backend-subnet**, Subnet address range=192.168.2.0/24, Leave NAT Gateway, Network security group and Route table as None → Save



Add subnet

Name *
Backend-subnet

Subnet address range * ⓘ
192.168.2.0/24
192.168.2.0 - 192.168.2.255 (251 + 5 Azu

□ Add IPv6 address space ⓘ

NAT gateway ⓘ
None

Network security group
None

Route table
None

SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources fr
over service endpoints. Learn more

Services ⓘ
0 selected

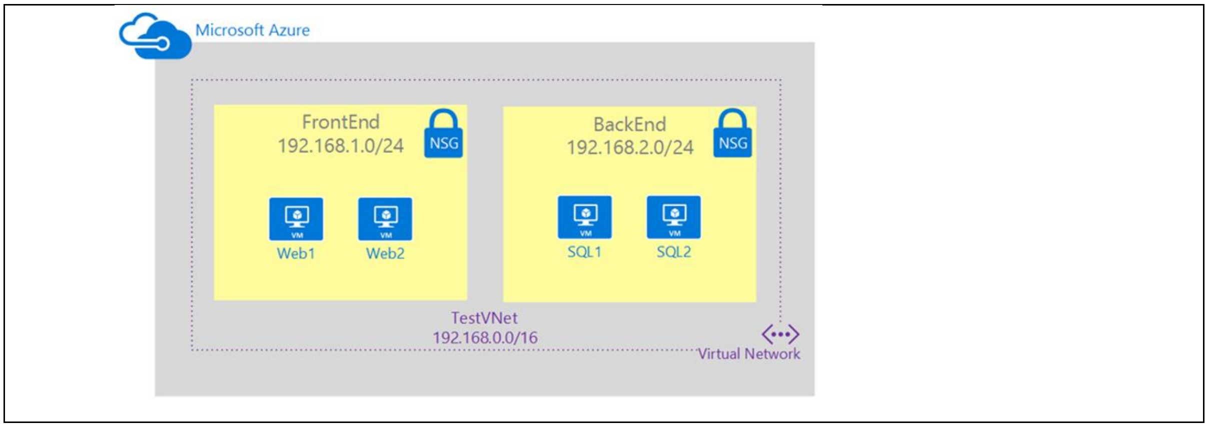
SUBNET DELEGATION

Network Security Group

NSGs are simple, stateful packet inspection devices that use the 5-tuple (the source IP, source port, destination IP, destination port, and layer 4 protocol) approach to create **allow/deny rules** for network traffic. You allow or deny traffic to and from a single IP address, to and from multiple IP addresses, or to and from entire subnets.

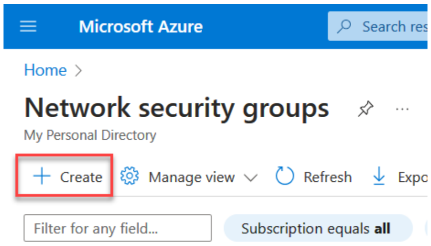
In this scenario you will create an NSG for each subnet in the **Demo-vnet** virtual network, as described below:

- **Frontend-nsg**. The front end NSG will be applied to the *FrontEnd* subnet, and contain two rules:
    - **rdp-allow**. This rule will allow RDP (3389) traffic to the *FrontEnd* subnet.
    - **web-allow**. This rule will allow HTTP (80) traffic to the *FrontEnd* subnet.
- **Backend-nsg**. The back end NSG will be applied to the *BackEnd* subnet, and contain two rules:
    - **sql-allow**. This rule allows SQL (1433) traffic only from the *FrontEnd* subnet.
    - **rdp-allow**: This rule will allow RDP (3389) traffic to the *BackEnd* subnet
    - **web-deny**. This is Outbound rule **denies all internet bound** traffic **from** the *BackEnd* subnet.



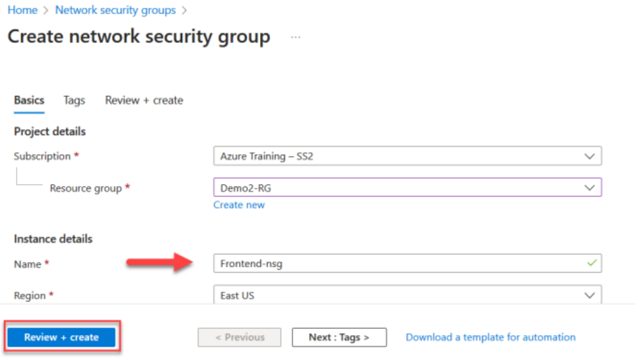
Create NSG for Frontend-subnet:

8. Search → Network Security Groups → + Create



9. Name=**Frontend-nsg** → Review + Create button

Go to resource after creation


Home > Network security groups >
Create network security group
Basics
Tags
Review + create
Review + create
< Previous
Next : Tags >
Download a template for automation

Project details

Subscription * Azure Training – SS2

Resource group * Demo2-RG

Create new

Instance details

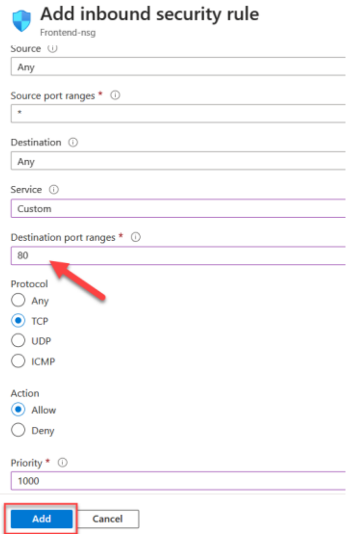
Name * Frontend-nsg

Region * East US

Create HTTP and RDP Rules

10. Select Frontend-nsg → Settings →

a. Inbound security rules → Add, Name=**AllowHTTP**, priority, Priority=1000, Source=Any, Source port range=*, Protocol=**TCP**, Destination=Any, Destination port range=**80**, Action=Allow → OK



Add inbound security rule
Frontend-nsg

Source: Any

Source port ranges *: *

Destination: Any

Service: Custom

Destination port ranges *: 80

Protocol
○ Any
● TCP
○ UDP
○ ICMP

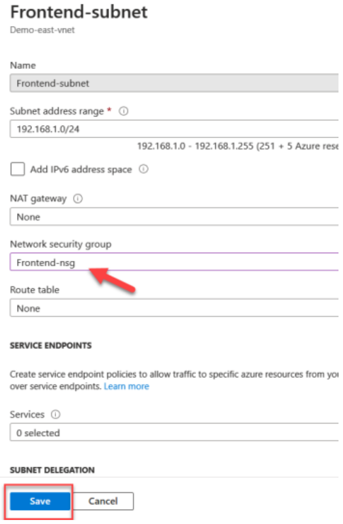
Action
● Allow
○ Deny

Priority *: 1000

b. Inbound security rules → Add, Name=**AllowRDP**, priority, Priority=1001, Source=Any, Source port range=*, Protocol=**TCP**, Destination=Any, Destination port range=**3389**, Action=Allow → OK

Associate the NSG to the FrontEnd subnet

11. Select Demo-east-vnet → Subnets → **Frontend-subnet** → Network security group → Select Frontend-nsg → Save



Frontend-subnet
Demo-east-vnet

Name: Frontend-subnet

Subnet address range *: 192.168.1.0/24
192.168.1.0 - 192.168.1.255 (251 + 5 Azure rese

□ Add IPv6 address space

NAT gateway: None

Network security group: Frontend-nsg

Route table: None

SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources from yo over service endpoints. Learn more

Services: 0 selected

SUBNET DELEGATION

12. **Create NSG for Backend:** Browse → Network Security Groups → Add → Name=Backend-nsg → Create

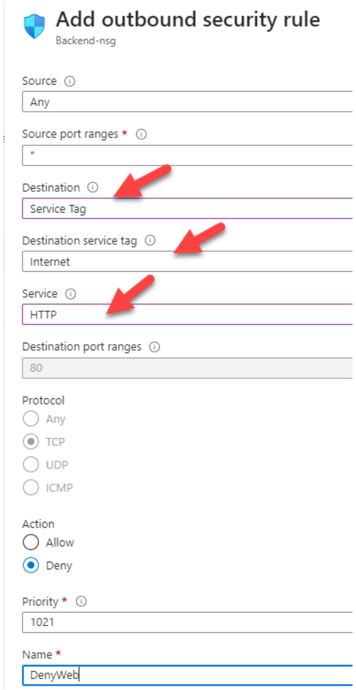
Configuring rules for Backend-subnet

13. Select Backend-nsg →

a. **Inbound security rules** → Add, Name=**AllowSQL**, priority, Priority=1001, Source=**CIDR block**, **Source IP address range=192.168.1.0/24,** Source port range=*, Protocol=**TCP**, Destination=Any, Destination port range=**1433**, Action=Allow → OK

b. **Inbound security rules** → Add, Name=**AllowRDP**, priority, Priority=1002, Source=Any, Source port range=*, Protocol=**TCP**, Destination=Any, Destination port range=3389, Action=Allow → OK

c. **Outbound security rules** → Add, Name=**DenyWeb**, priority, Priority=1000, Destination=**Tag**, destination Tag=**Internet**, Destination port range=80, Source=**Any**, Protocol=**Any**, Source port range=*, Action=**Deny** → OK



Add outbound security rule
Backend-nsg

Source: Any

Source port ranges *: *

Destination: Service Tag

Destination service tag: Internet

Service: HTTP

Destination port ranges: 80

Protocol
○ Any
● TCP
○ UDP
○ ICMP

Action
○ Allow
● Deny

Priority *: 1021

Name *: DenyWeb

Associate the NSG to the BackEnd subnet

14. Select Demo-eastus-vnet → Subnets → Backend-subnet → Network security group → Select Backend-nsg → Save

Summary:

Virtual Network (192.168.0.0/16)

Frontend-subnet (192.168.1.0/24)

Frontend-nsg

Allow RDP / HTTP (Inbound)

Backend-subnet (192.168.2.0/24)

Backend-nsg

Allow RDP / SQL (Inbound)

Deny Internet (Outbound)

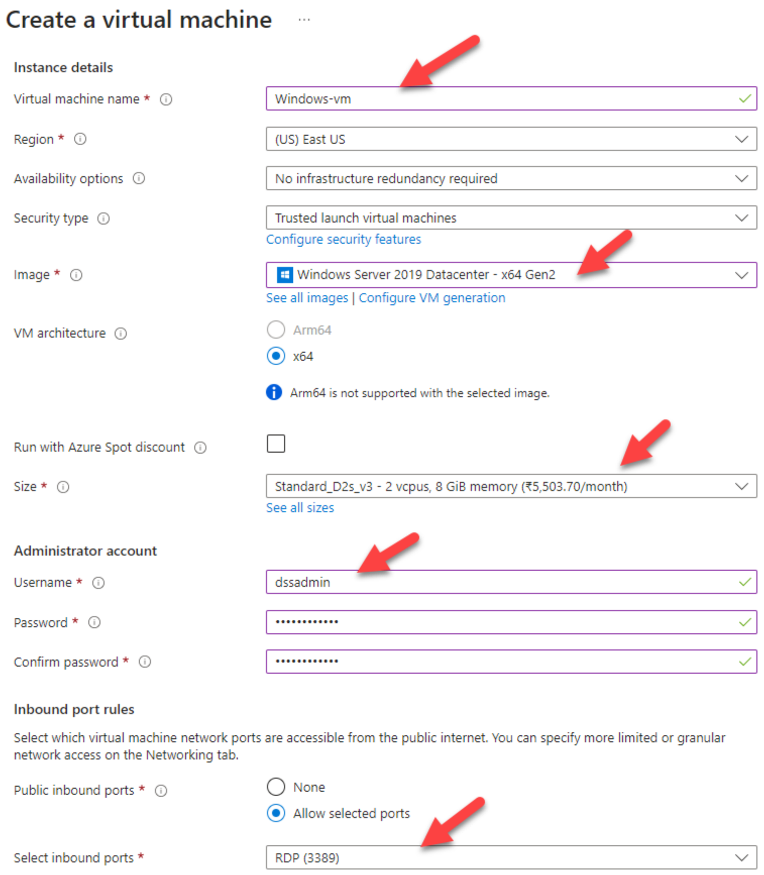
Creating a Virtual Machine

15. Azure portal → Search → Compute → Virtual machines

16. Click on + Create → Azure virtual machine

Basics tab →

- Select same Subscription, Resource group, Region as that of Virtual network created above.
- Virtual machine name = Web1-vm,
- Image = "Windows Server 2019 Datacenter x64 Gen 2"
- Username = dssadmin,
- Password = Password@123,
- Public Inbound Port = None.



Create a virtual machine ...

Instance details

Virtual machine name * : Windows-vm

Region * : (US) East US

Availability options : No infrastructure redundancy required

Security type : Trusted launch virtual machines

Configure security features

Image * : Windows Server 2019 Datacenter - x64 Gen2

See all images | Configure VM generation

VM architecture : ○ Arm64 ● x64

Arm64 is not supported with the selected image.

Run with Azure Spot discount : □

Size * : Standard_D2s_v3 - 2 vcpus, 8 GiB memory (₹5,503.70/month)

See all sizes

Administrator account

Username * : dssadmin

Password * : ••••••••••••

Confirm password * : ••••••••••••

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * : ○ None ● Allow selected ports

Select inbound ports * : RDP (3389)

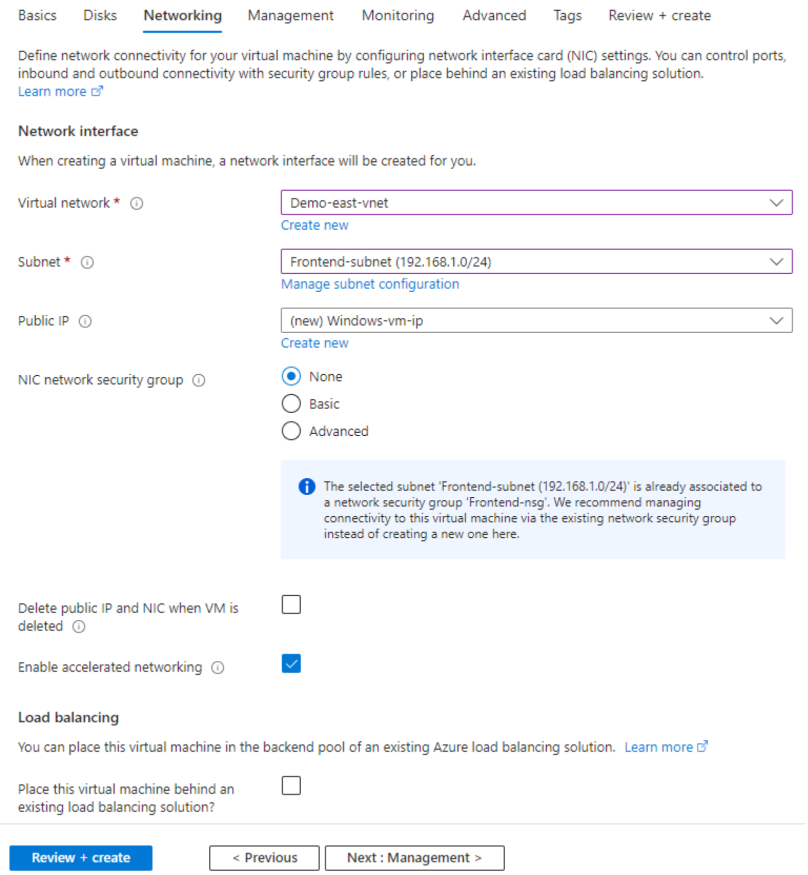
Disks tab → All Defaults

Networking tab →

Virtual Network = Select the earlier created VNET,

Subnet=Frontend-subnet

NSG = None



Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. Learn more

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * : Demo-east-vnet

Create new

Subnet * : Frontend-subnet (192.168.1.0/24)

Manage subnet configuration

Public IP : (new) Windows-vm-ip

Create new

NIC network security group : ● None ○ Basic ○ Advanced

The selected subnet 'Frontend-subnet (192.168.1.0/24)' is already associated to a network security group 'Frontend-nsg'. We recommend managing connectivity to this virtual machine via the existing network security group instead of creating a new one here.

Delete public IP and NIC when VM is deleted □

Enable accelerated networking ☑

Load balancing

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. Learn more

Place this virtual machine behind an existing load balancing solution? □

Leave the other tabs as is.

Click **Review + Create** → Review your configuration and Click **Create**

17. Create another VM (All steps same as Web1-vm Except)
    a. Name=Database1-vm
    b. Subnet = Backend-subnet

Summary:

Demo-eastus-vnet

Frontend-subnet

Frontend-nsg

Allowed HTTP and RDP

Web1-vm

NO NSG (NIC Level)

Remote Login and installed IIS

edit wwwroot\iisstart.png

Web1-vm-ip

DNS Name

Backend-sub

Backend-nsg

Allowed RDP Inbound

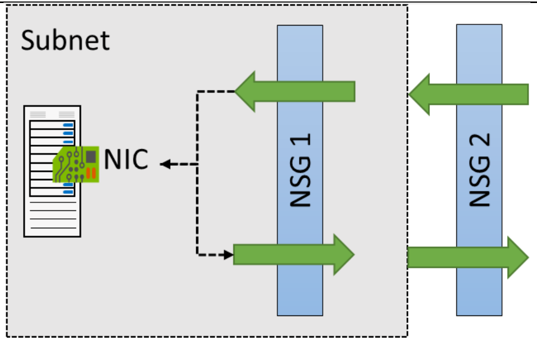
Denied Internet: OutBound

Database1-vm

NO NSG (NIC Level)

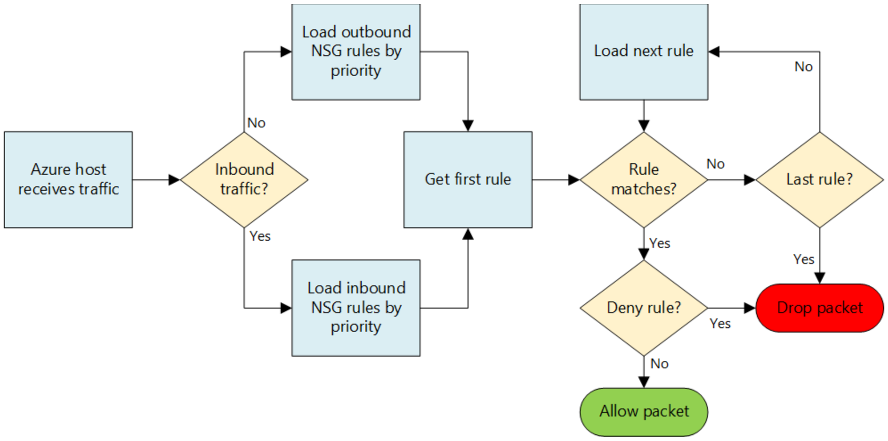
NSG: Evaluate effective security rules

Be very careful when you want to apply NSG to both VM (NIC) and subnet level at the same time. NSGs are evaluated independently, and an “allow” rule must exist at **both levels** otherwise traffic will not be admitted.



If there was incoming traffic on port 80, you would need to have the NSG at subnet level ALLOW port 80, and you would also need another NSG with ALLOW rule on port 80 at the NIC level. For incoming traffic, the NSG set at the subnet level is evaluated first, then the NSG set at the NIC level is evaluated. For outgoing traffic, it is the converse.

The picture below should even clarify this concept more: you can see how rules are evaluated for network packets, once again remember that you need to **evaluate this diagram two times**: once for subnet level NSG rules, and once for NIC level NSG rules.

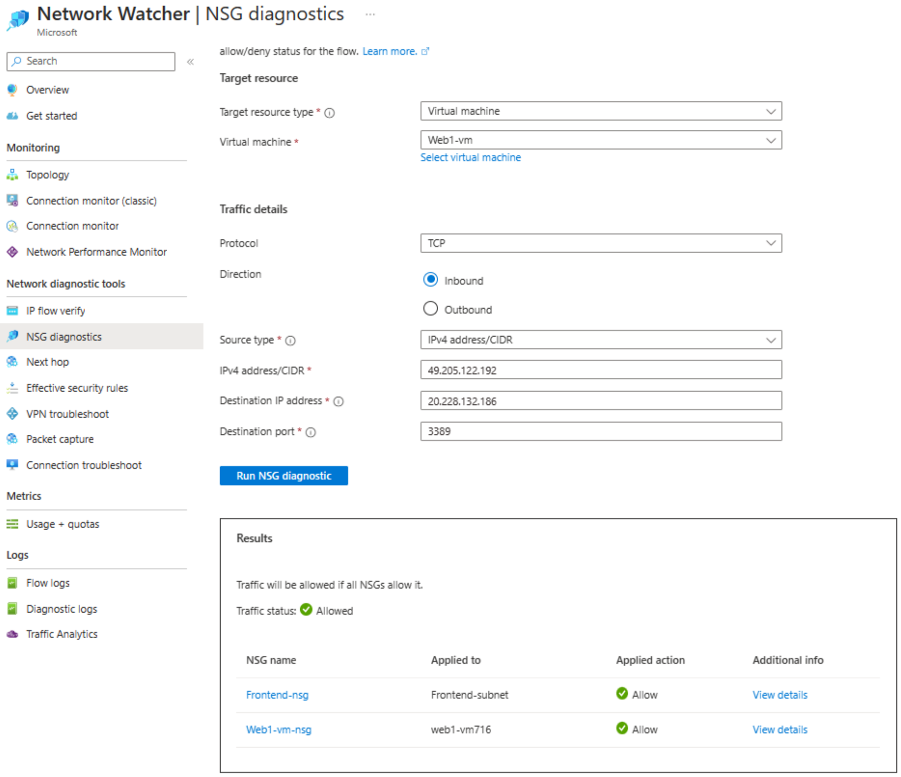


To see the Effective Rules:

Select the VM → Settings → Networking → Click on **Effective security rules**

Now you get an overview which NSGs are associated with the VM's NIC and which rules are applied to it.

For an offline analysis there is a download option, that generates a CSV file of the output.



allow/deny status for the flow. Learn more.

Target resource

Target resource type *	Virtual machine
Virtual machine *	Web1-vm Select virtual machine

Traffic details

Protocol	TCP
Direction	● Inbound ○ Outbound
Source type *	IPv4 address/CIDR
IPv4 address/CIDR *	49.205.122.192
Destination IP address *	20.228.132.186
Destination port *	3389

Results

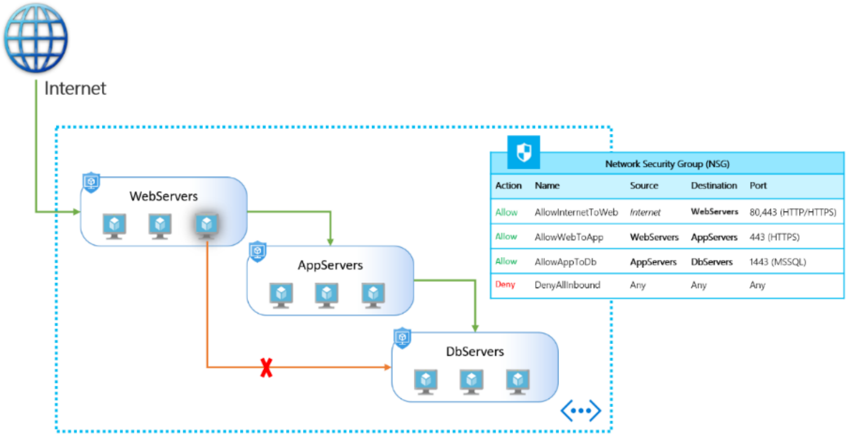
Traffic will be allowed if all NSGs allow it.

Traffic status: ✓ Allowed

NSG name	Applied to	Applied action	Additional info
Frontend-nsg	Frontend-subnet	✓ Allow	View details
Web1-vm-nsg	web1-vm716	✓ Allow	View details

Application Security Groups

Application security groups enable you to configure network security as a natural extension of an application's structure, allowing you to group virtual machines and define network security policies based on those groups. This feature allows you to reuse your security policy at scale without manual maintenance of explicit IP addresses. The platform handles the complexity of explicit IP addresses and multiple rule sets, allowing you to focus on your business logic.



Network Security Group (NSG)

Action	Name	Source	Destination	Port
Allow	AllowInternetToWeb	*Internet*	**WebServers**	80,443 (HTTP/HTTPS)
Allow	AllowWebToApp	**WebServers**	**AppServers**	443 (HTTPS)
Allow	AllowAppToDb	**AppServers**	**DbServers**	1443 (MSSQL)
Deny	DenyAllInbound	Any	Any	Any

Create two new Application Secuirty Groups

a) WebServers-asg

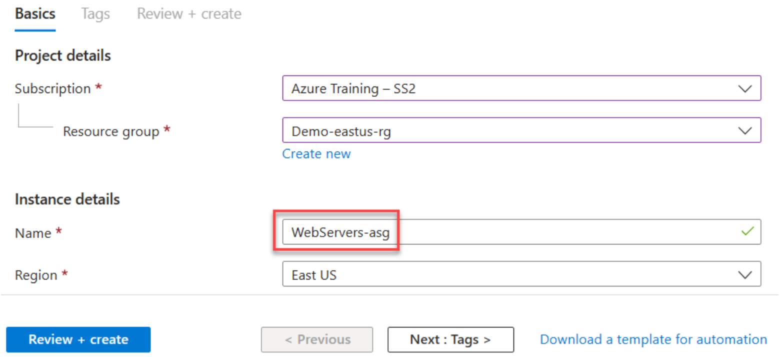
b) DbServers-asg

18. Search → Application Security Groups → + Create

Name=**WebServers-asg**

Region = East US

Review + Create



Project details

Subscription *	Azure Training – SS2
Resource group *	Demo-eastus-rg

Create new

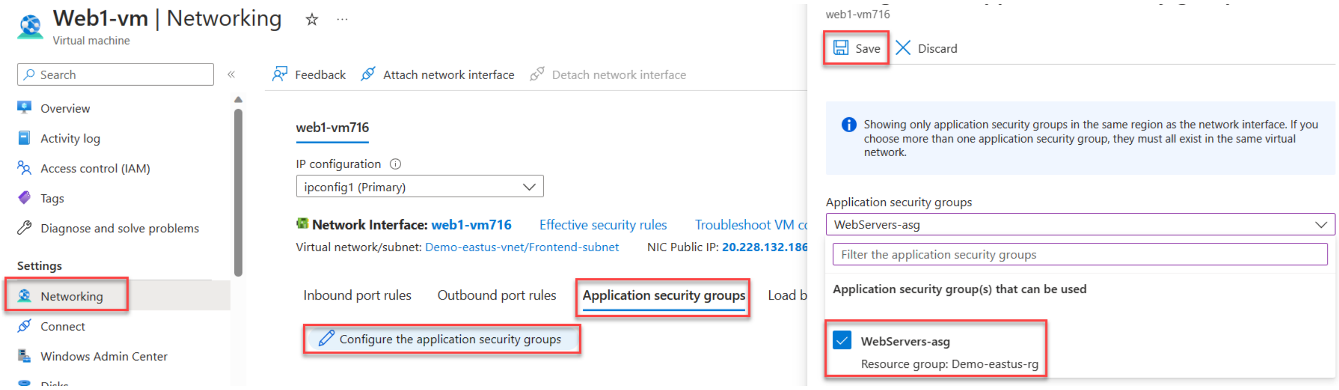
Instance details

Name *	WebServers-asg
Region *	East US

19. Similarly create another Application security group = **DatabaseServers-asg**

Attach them to respective VM:

20. Web1-vm → Networking → **Application Securty Group** tab → click on Configure the application security groups → Select WebServers-asp → Save



IP configuration: ipconfig1 (Primary)

Network Interface: web1-vm716

Virtual network/subnet: Demo-eastus-vnet/Frontend-subnet NIC Public IP: 20.228.132.186

Showing only application security groups in the same region as the network interface. If you choose more than one application security group, they must all exist in the same virtual network.

Application security groups: WebServers-asg

Filter the application security groups

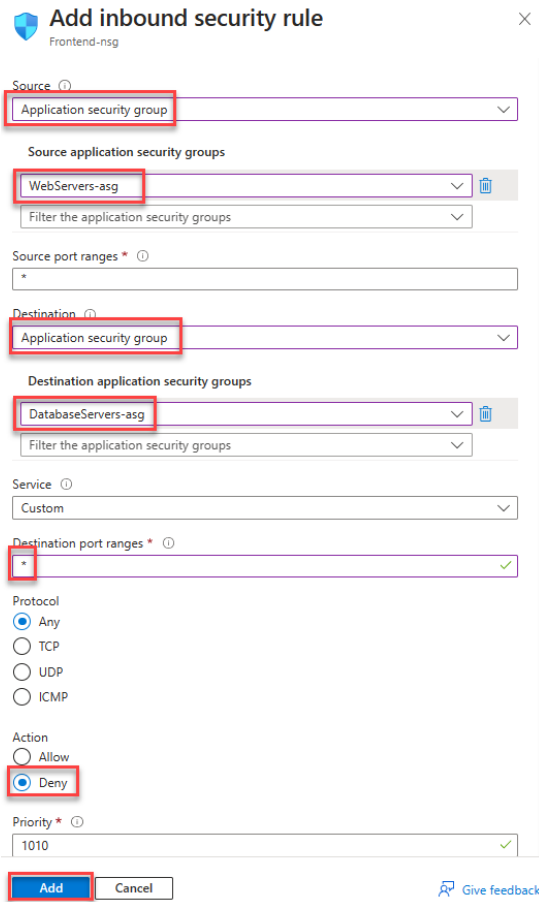
☑ WebServers-asg
Resource group: Demo-eastus-rg

21. Database1-vm → Networking → **Application Securty Group** tab → click on Configure the application security groups → Select DatabaseServers-asp → Save

Create NSG rule to block traffic from all VMs with ASG=WebServers-asg to all VMs with ASG=DatabaseServers-asg

22. Select **Frontend-nsg → Outbound security rules** → Add

23. Select values as in image below:



Add inbound security rule

Frontend-nsg

Source: Application security group

Source application security groups: WebServers-asg

Filter the application security groups

Source port ranges *: *

Destination: Application security group

Destination application security groups: DatabaseServers-asg

Filter the application security groups

Service: Custom

Destination port ranges *: *

Protocol

● Any

○ TCP

○ UDP

○ ICMP

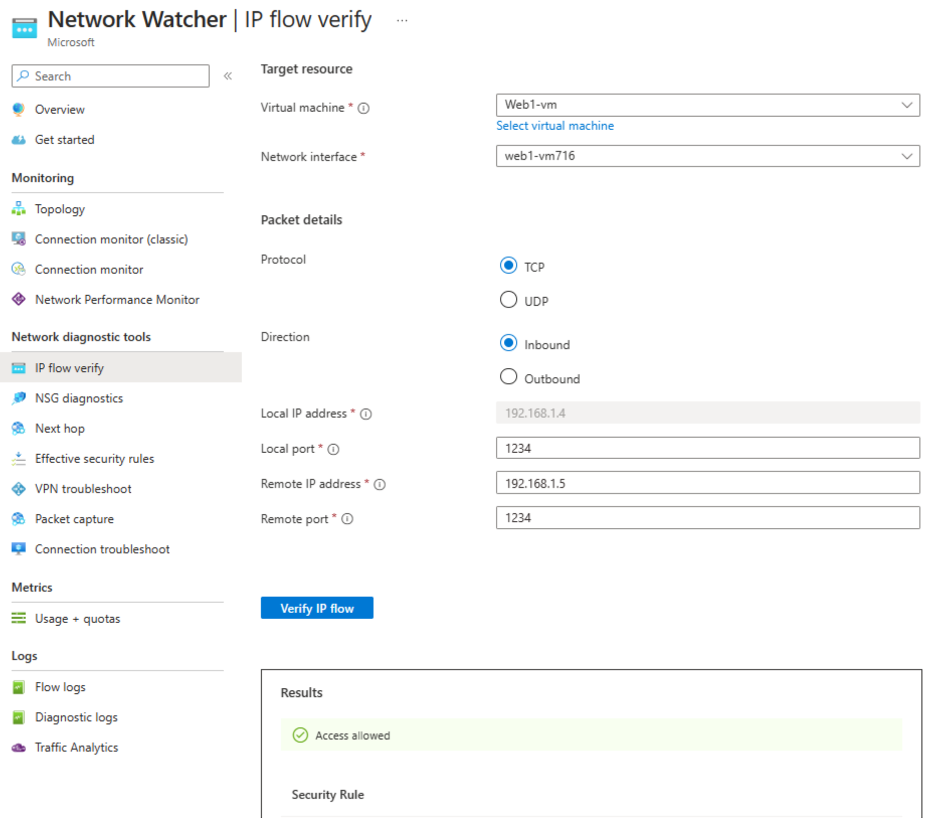
Action

○ Allow

● Deny

Priority *: 1010

24. Wait for couple of minutes.
25. Use **Network Watcher** and note that **IP Flow verify** is **failed** from Web1-vm to Database1-vm
26. Search Network Watcher → IP flow verify



Target resource

Virtual machine * ⓘ: Web1-vm
Select virtual machine

Network interface *: web1-vm716

Packet details

Protocol: ● TCP ○ UDP

Direction: ● Inbound ○ Outbound

Local IP address * ⓘ: 192.168.1.4

Local port * ⓘ: 1234

Remote IP address * ⓘ: 192.168.1.5

Remote port * ⓘ: 1234

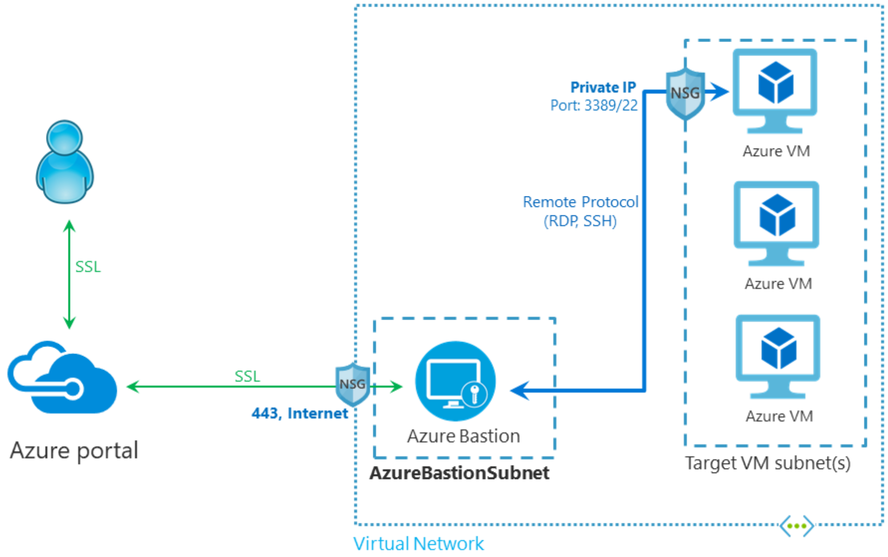
Results

✓ Access allowed

Security Rule

Azure Bastion

Azure Bastion is a fully managed PaaS service that provides secure and seamless RDP and SSH access to your virtual machines directly through the Azure Portal. Azure Bastion is provisioned directly in your Virtual Network (VNet) and supports all VMs in your Virtual Network (VNet) using SSL without any exposure through public IP addresses.



In this diagram:

- The Bastion host is deployed in the virtual network and in subnet **AzureBastionSubnet**.
- The user connects to the Azure portal using any **HTML5** browser.
- The user selects the virtual machine to **connect** to.
- With a single click, the **RDP/SSH session** opens in the browser.
- No public IP is required on the Azure VM.

The following features are available:

- **RDP and SSH directly in Azure portal:** You can directly get to the RDP and SSH session directly in the Azure portal using a single click seamless experience.
- **Remote Session over SSL and firewall traversal for RDP/SSH:** Azure Bastion uses an HTML5 based web client that is automatically streamed to your local device, so that you get your RDP/SSH session over SSL on port 443 enabling you to traverse corporate firewalls securely.
- **No Public IP required on the Azure VM:** Azure Bastion opens the RDP/SSH connection to your Azure virtual machine using private IP on your VM. You don't need a public IP on your virtual machine.
- **No hassle of managing NSGs:** Azure Bastion is a fully managed platform PaaS service from Azure that is hardened internally to provide you secure RDP/SSH connectivity. You don't need to apply any NSGs on Azure Bastion subnet. Because Azure Bastion connects to your virtual machines over private IP, you can configure your NSGs to allow RDP/SSH from Azure Bastion only. This removes the hassle of managing NSGs each time you need to securely connect to your virtual machines.

- Azure Bastion can support up to **25 concurrent RDP**, this is still dependent on the Azure Virtual Machines. Azure Virtual Machine doesn't support more than 2 concurrent RDP connections and these must be from two different user accounts.
- Each instance can support 20 concurrent RDP connections and 40 concurrent SSH connections for medium workloads.

Create a bastion host

1. Select Demo-eastus-vnet → Subnets → + Subnet
    a. Name **AzureBastionSubnet** (You must use a subnet of at least /26 or larger eg: /26, /25 and …)
    b. Subnet address range = 192.168.3.0/24
2. Search Bastions → + Create
    a. Name = **Demo-eastus-bastion**
    b. Region = eastus
    c. Tier = Basic
    d. Virtual network = Demo-eastus-vnet
    e. Subnet = AzureBastionSubnet
    f. Public IP address = Create new
    g. Public IP address name = Demo-eastus-bastion-ip

OR you can use the Shortcut:

Azure Virtual Network → Demo-eastus-vnet → Bastion → Deploy Bastion

Connect to VM

3. Azure VM → **Connect** → **Bastion** tab → Use Bastion → Provide the RDP Username and Password → Connect

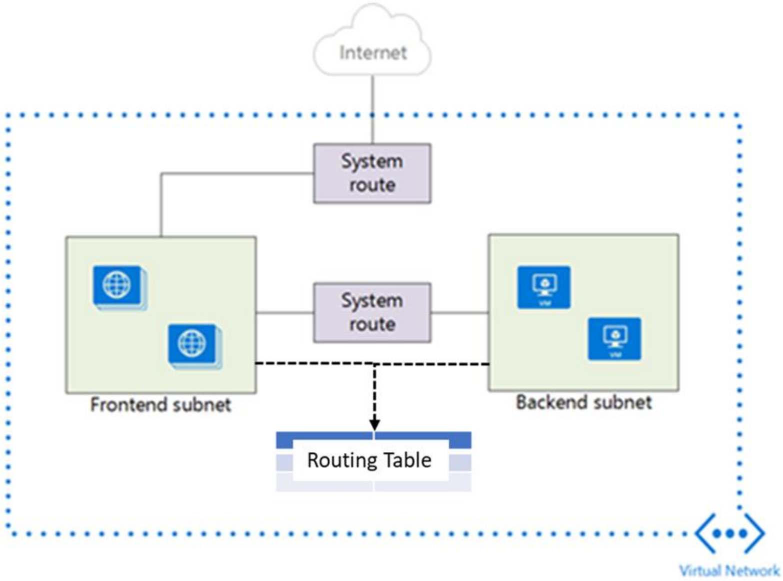
User Defined Route Table

- When you add virtual machines (VMs) to a virtual network (VNet) in Azure, you will notice that the VMs are able to communicate with each other over the network, automatically. You do not need to specify a gateway, even though the VMs are in different subnets. The same is true for communication from the VMs to the public Internet, and even to your on-premises network when a hybrid connection from Azure to your own datacenter is present.

- This flow of communication is possible because Azure uses a series of **system routes** to define how IP traffic flows.

System routes control the flow of communication in the following scenarios:

- From within the same subnet.
- From a subnet to another within a VNet.
- From VMs to the Internet.
- From a VNet to another VNet through a VPN gateway.
- From a VNet to another VNet through VNet Peering (Service Chaining).
- From a VNet to your on-premises network through a VPN gateway.



Information about the **system routes** is recorded in a **route table**. A route table contains a set of **rules**, called **routes**, that specifies how packets should be routed in a virtual network. Route tables are **associated to subnet**s, and each packet leaving a subnet is handled based on the associated route table. Packets are matched to routes using the destination. The destination can be an **IP address, a virtual network gateway, a virtual appliance, or the internet**. If a matching route can't be found, then the packet is **dropped.**

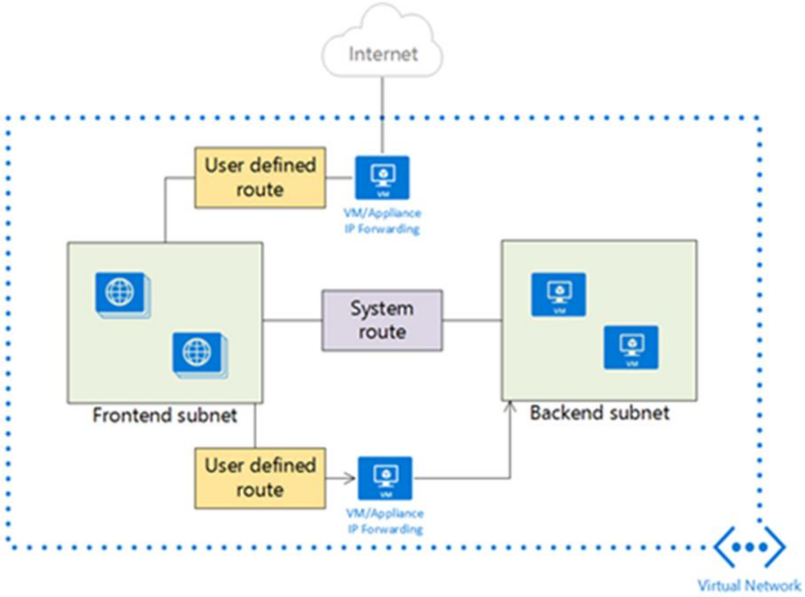
User Defined Routes

For most environments you will only need the system routes already defined by Azure.

However, you may need to create a route table and add one or more routes in specific cases, such as:

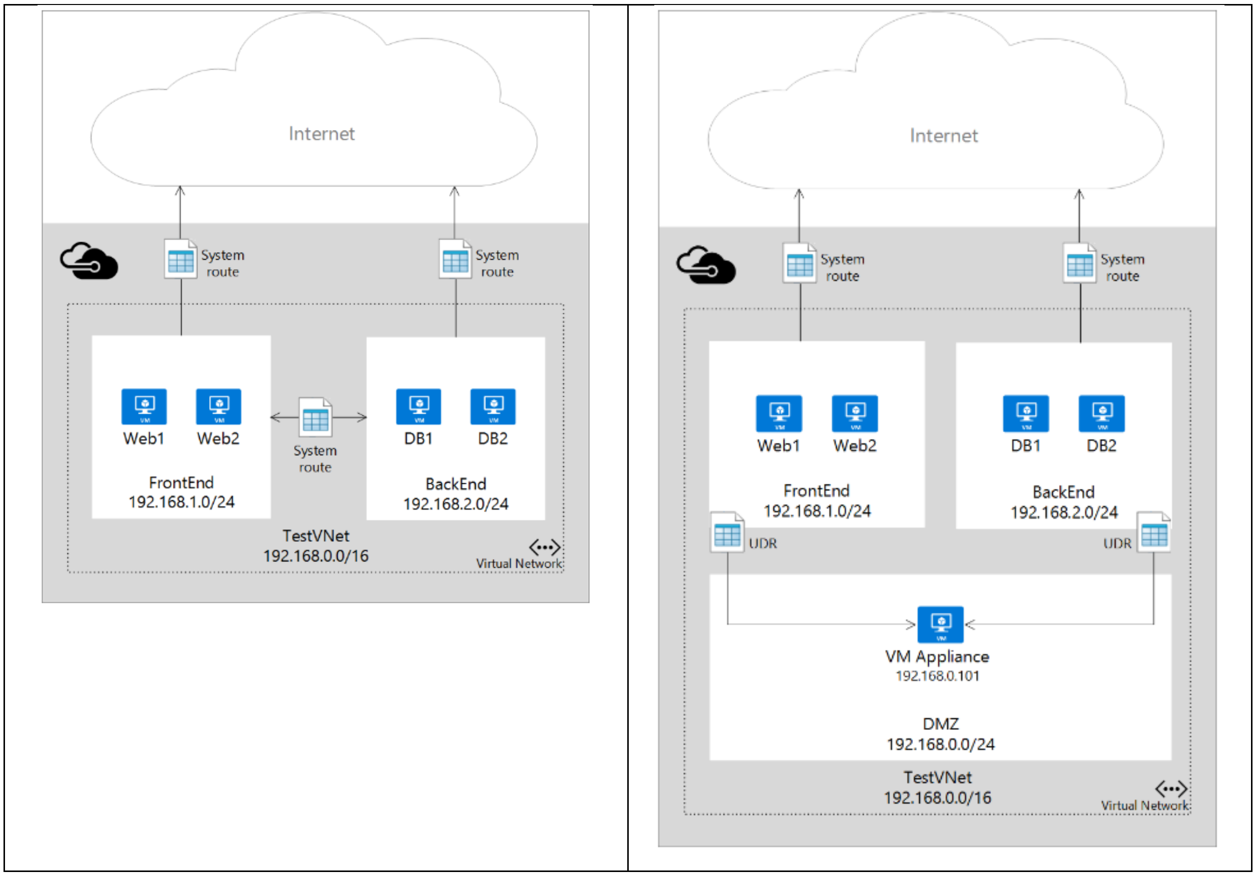
- ○ Use of virtual appliances in your Azure environment.

- Force tunneling to the Internet via your on-premises network.



Each route table can be associated to multiple subnets, but a subnet can only be associated to a single route table.

There are no additional charges for creating route tables in Microsoft Azure.



- User defined routes are only applied to **traffic leaving a subnet**. You cannot create routes to specify how traffic comes into a subnet from the Internet, for instance. Also, the appliance you are forwarding traffic to cannot be in the same subnet where the traffic originates. **Always create a separate subnet for your appliances**.
- NVAs are VMs that help with network functions like routing and firewall optimization. Some of the cases where virtual appliances can be used include:
    - Monitoring traffic with an intrusion detection system (IDS).
    - Controlling traffic with a firewall.
- This **virtual appliance VM** must be able to receive incoming traffic that is not addressed to itself. To allow a VM to receive traffic addressed to other destinations, you must **enable IP Forwarding** for the VM. This is an Azure setting, not a setting in the guest operating system.
- You can have multiple route tables, and the same route table can be associated to one or more subnets. And each subnet can only be associated to a single route table.

NOTE: An **intrusion detection system** (IDS) is a device or software application that monitors a network or systems for **malicious activity or policy violations**. Any malicious activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources, and uses alarm filtering techniques to distinguish malicious activity from false alarms.

The most common classifications are **network intrusion detection systems** (**NIDS**) and **host-based intrusion detection systems** (HIDS).

Network security capabilities of virtual network security appliances include:

- Firewalling
- Intrusion detection/intrusion prevention
- Vulnerability management
- Application control
- Network-based anomaly detection
- Web filtering
- Antivirus
- Botnet protection

To find available Azure virtual network security appliances, go to the Azure Marketplace and search for “security” and “network security.”

Example of Virtual Appliance: Palo Alto Networks VM-Series.

Create User Defined Routes (UDR) :

1. Select Demo-eastus-vnet → Subnets → + Subnet
    a) **Name = VirtualAppliance-subnet**
    b) Subnet address range = 192.168.4.0/24.
2. Create a **new VM** (Demo-va) to be used for virtual appliance with private IP address **192.168.4.4** (preferably static ip)
    a) Basic Tab:
        - Virtual machine name = Demo-va
        - Region = East US
        - Image = Windows Server 2022 Datacenter

    b) Networking

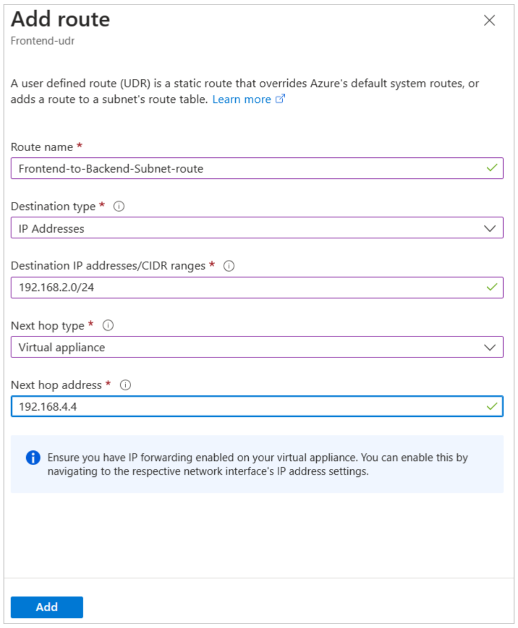
- Virtual network = Demo-eastus-net
- Subnet = **VirtualAppliance-subnet**

UDR for Frontend Subnet when target is any VM in backend subnet

3. Create UDR: Search Bar → **Route tables** → + Create
   a) Region = East US
   b) Name=**Frontend-udr**
   c) Virtual network gateway route propagation = Enabled (default)
   d) Review + Create → Create

Border Gateway Protocol (**BGP**): An on-premises network gateway can exchange routes with an Azure virtual network gateway using the BGP. Routes are automatically added to the route table of all subnets with BGP propagation enabled.

4. Select Route table Frontend-udr → Routes → + Create
   a) Set Name=**Frontend-to-Backend-Subnet-route**,
   b) Destination type = IP Addresses
   c) Destination IP addresses/CIDR ranges = 192.168.2.0/24 (Range of Backend Subnet),
   d) Next hop type=**Virtual appliance**,
   e) Next hop address = **192.168.4.4** (Private IP of VM Appliance - Demo-va)
   f) Add



Add route
Frontend-udr

A user defined route (UDR) is a static route that overrides Azure's default system routes, or adds a route to a subnet's route table. Learn more

Route name * : Frontend-to-Backend-Subnet-route

Destination type * : IP Addresses

Destination IP addresses/CIDR ranges * : 192.168.2.0/24

Next hop type * : Virtual appliance

Next hop address * : 192.168.4.4

Ensure you have IP forwarding enabled on your virtual appliance. You can enable this by navigating to the respective network interface's IP address settings.

Routing Algorithms:

a) If multiple routes contain the **same address prefix**, Azure selects the route type, based on the following priority:

1. User-defined route
2. BGP route
3. System route

b) Longest prefix match algorithm

For example, if the destination address is 10.0.0.5 and there are two routes: One route specifies the 10.0.0.0/24 address prefix, while the other route specifies the 10.0.0.0/16 address prefix. In this case, Azure selects a route using the longest prefix match algorithm, which is the 10.0.0.0/24 route.

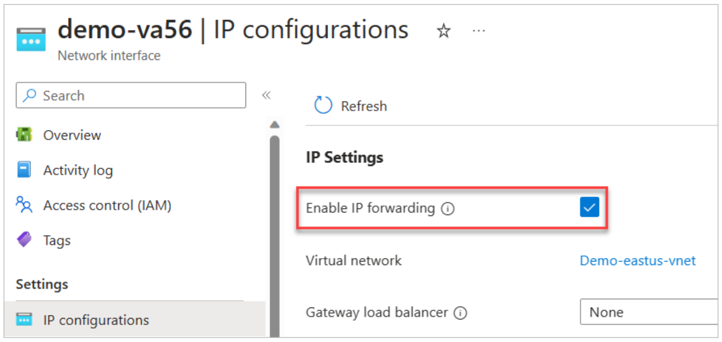
c) A route with the **0.0.0.0/0** address prefix instructs Azure how to route traffic destined for an IP address that is not within the address prefix of any other route in a subnet's route table.

To Associate the Route Table with Subnet

5. Select Frontend-udr-table → Subnets → **+Associate** → Select **Frontend-subnet**

For the VM in New Subnet (used for Virtual Appliance):

6. Enable IP Forwarding for NIC of **Demo-va** VM.
   a) Demo-va → **Networking** → Click on **Network Interface Card** (eg: **demo-vaXX)** → IP Configurations → **IP Forwarding = Enable**



IP Settings

Enable IP forwarding ☑

Virtual network: Demo-eastus-vnet

Gateway load balancer: None

7. Turn on **IP forwarding** for Demo-va (**Virtual Appliance VM)** at Operating System leve.
   a) RDP to Virtual Appliance VM (Demo-va) →
   b) Search and Open **PowerShell** Console
   c) Execute the following command in the console window

```
Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters -Name
IpEnableRouter -Value 1
```

d) **Restart the Virtual Appliance VM: Azure Portal → Demo-va → Restart.**

8. In Database1-vm (with PrivateIP 192.168.2.4), Enable Internet Control Message Protocol (ICPM) which the Windows Firewall denies by default.
   a) RDP to Database1-vm (In Backend subnet)
   b) Search and Open **PowerShell** Console in **Administrator** mode
   c) Execute the command on VM's

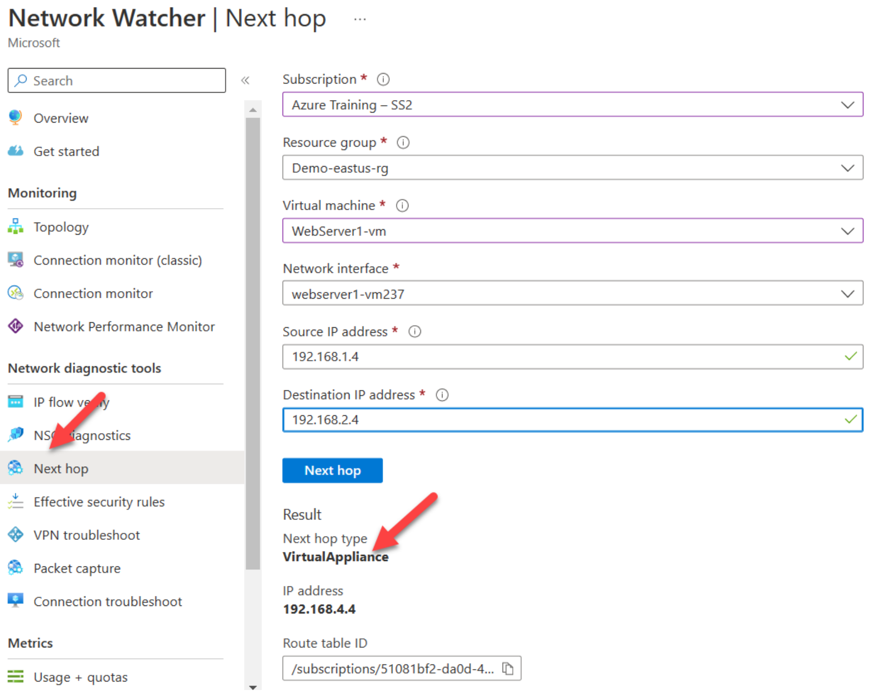
```
New-NetFirewallRule -DisplayName "Allow ICMPv4-In" -Protocol ICMPv4
```

9. Test the routing of network traffic
   a) RDP to Web1-vm (Frontend subnet)
   b) Search and Open **PowerShell** Console in **Administrator** mode
   c) Execute the following command

```
tracert  192.168.2.4 <Target VM Name from Backend-subnet>
```

Note that the first hop is Virtual Appliance VM and send hop to the target VM

OR Use Network Watcher



Network Watcher | Next hop

Microsoft

Subscription *: Azure Training – SS2

Resource group *: Demo-eastus-rg

Virtual machine *: WebServer1-vm

Network interface *: webserver1-vm237

Source IP address *: 192.168.1.4

Destination IP address *: 192.168.2.4

Result

Next hop type: **VirtualAppliance**

IP address: **192.168.4.4**

Route table ID: /subscriptions/51081bf2-da0d-4...

Rules Explained:

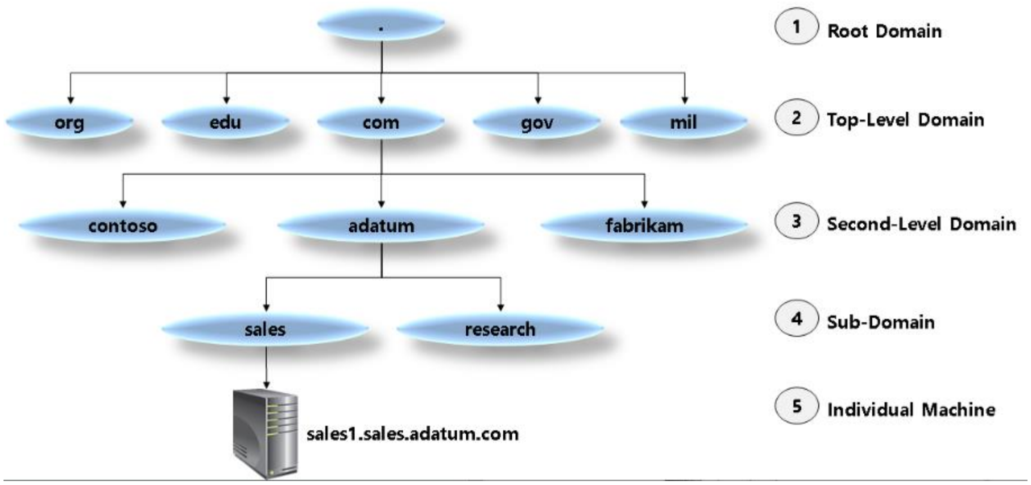
https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview

Azure DNS

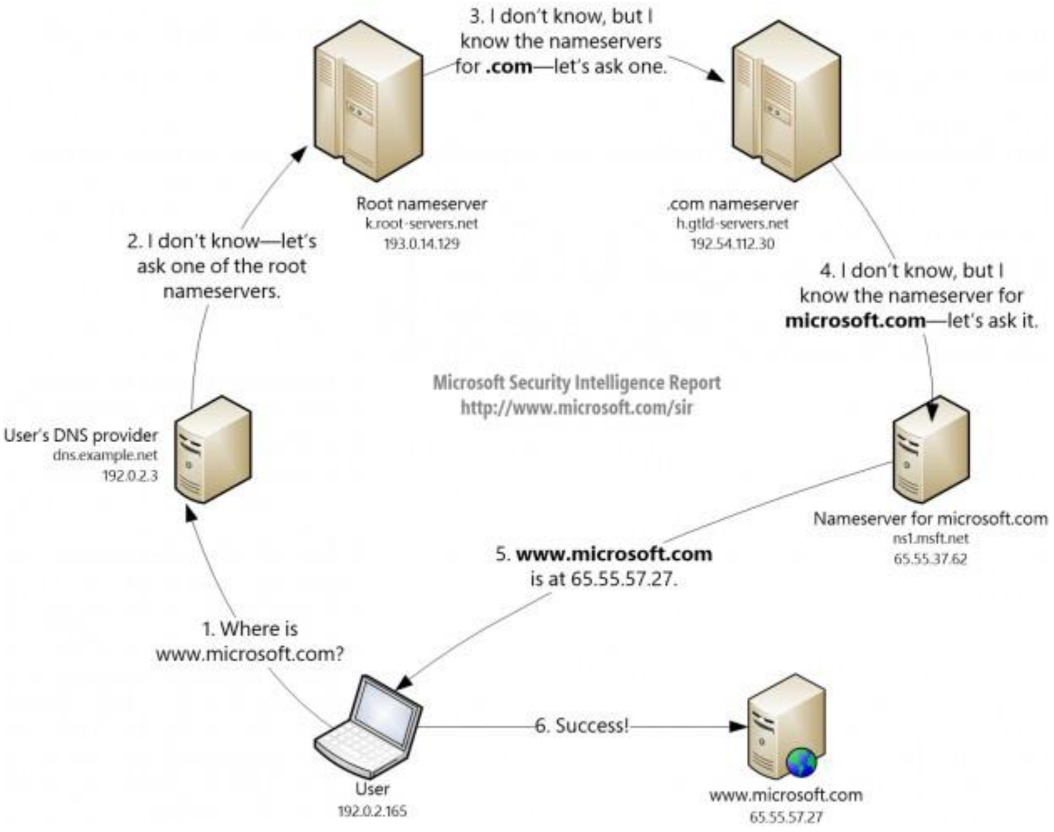
- The Domain Name Service, or DNS, is responsible for translating (or resolving) a website or service name to its IP address.
- Azure DNS is a hosting service for DNS domains, providing name resolution using Microsoft Azure infrastructure.
- Applications requiring automatic DNS management can integrate with the service via the REST API and SDKs.
- When you add a new DNS record, the Azure DNS name servers are updated in a few seconds so you don't have to wait long before that DNS record can be used.
- Azure DNS does not currently support purchasing of domain names.

DNS Domains:

The DNS is a hierarchy of domains. The hierarchy starts from the 'root' domain, whose name is simply '.'. Below this come top-level domains, such as 'com', 'net', 'org', 'uk' or 'jp'. Below these are second-level domains, such as 'org.uk' or 'co.jp'. The domains in the DNS hierarchy are globally distributed, hosted by DNS name servers around the world.



DNS Resolution: To answer queries, it uses aspecial type of DNS record called a Name Server (NS)record. For example, the root zone contains NS records for ‘com’and shows the name servers for the ‘com’ zone. In turn,the ‘com’ zone contains NS records for ‘contoso.com’, whichshows the name servers for the ‘contoso.com’ zone. Settingup the NS records is called delegating the domain.


3. I don't know, but I know the nameservers for .com—let's ask one.
Root nameserver
k.root-servers.net
193.0.14.129
.com nameserver
h.gtld-servers.net
192.54.112.30
2. I don't know—let's ask one of the root nameservers.
4. I don't know, but I know the nameserver for microsoft.com—let's ask it.
Microsoft Security Intelligence Report
http://www.microsoft.com/sir
User's DNS provider
dns.example.net
192.0.2.3
Nameserver for microsoft.com
ns1.msft.net
65.55.37.62
5. www.microsoft.com is at 65.55.57.27.
1. Where is www.microsoft.com?
6. Success!
User
192.0.2.165
www.microsoft.com
65.55.57.27

How DNS Server Works

In browser http://www.bestazuretraining.com

1. Browser will send request to DNS Server as configured in your machine for finding IP of www.bestazuretraining.com
2. DNS if has IP - It immediately returns
3. DNS does'nt have IP - It will send the request to ROOT Name Server
4. Root Name Server will query -> .com Name Server
5. .com Name Server will sent the request bestazuretraining.com name server
6. In bestazuretraining.com Name Server it will search for the required Recordset and return the value...
7. If IP is returned browser will directly send the request to target machine...
8. If Alias (CName) is returned then it again starts from Step 2...

<u>DNS Zone:</u>

A DNS zone is used to host the DNS records for a particular domain (purchased from registrar like godaddy.com). In order to start hosting your domain, you need to create a DNS zone. Any DNS record created for a particular domain will be inside a DNS zone for the domain.

For example, the domain "contoso.com" may contain a number of DNS records, such as "mail.contoso.com" (for a mail server) and "www.contoso.com" (for a web site).

Steps to Create a DNS Zone and Map Name to IP Address:

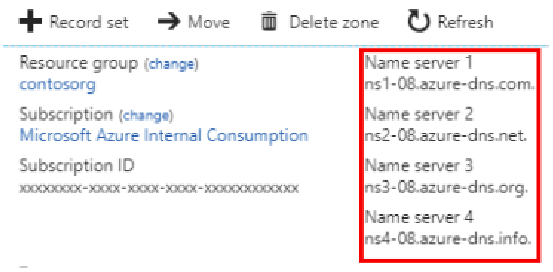
1. Buy a Domain Name from a Registrar (eg: godaddy.com is registrar)

2. Azure Portal → New → Networking → **DNS zone**

3. Name = deccansoft.net, Provide other details → Create

4. Goto Registrar Website → Login

5. DNS Delegation: Map domain Name Server to NS records for the DNS Zone created



Resource group (change)
contosorg

Subscription (change)
Microsoft Azure Internal Consumption

Subscription ID
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Name server 1
ns1-08.azure-dns.com.

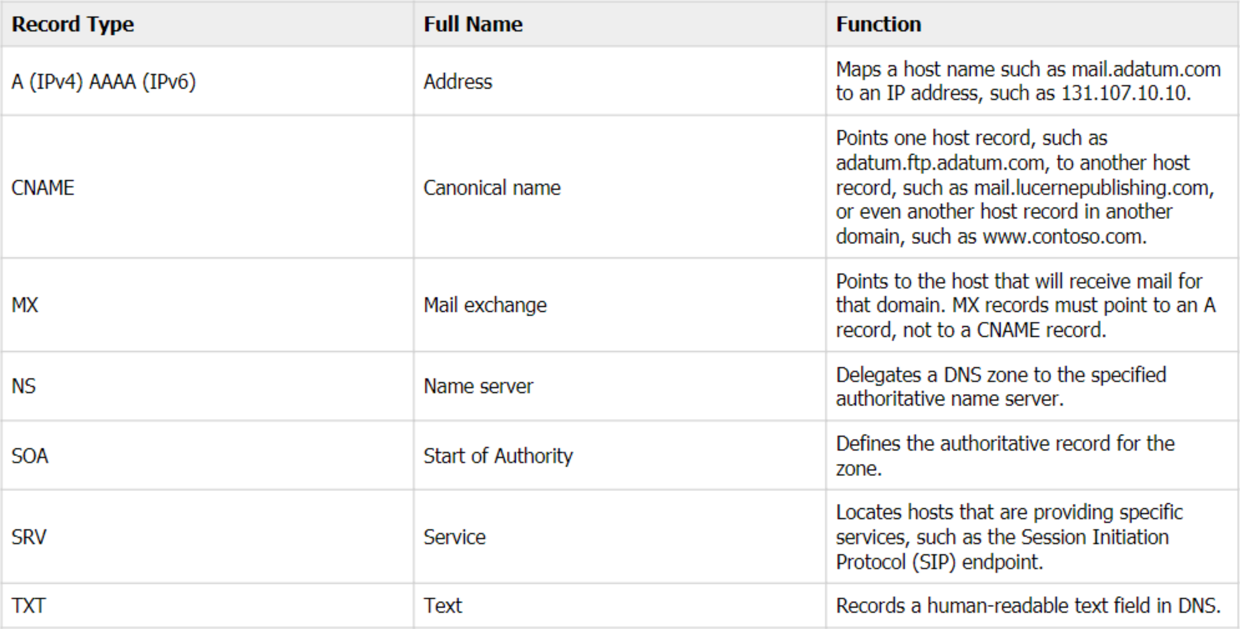
Name server 2
ns2-08.azure-dns.net.

Name server 3
ns3-08.azure-dns.org.

Name server 4
ns4-08.azure-dns.info.

6. Select the DNS Zone → **+ Record set** → Name=www, Type="A", TTL=1, IP Address=<Public IP of VM Created> → OK

DNS Record Type:



Record Type	Full Name	Function
A (IPv4) AAAA (IPv6)	Address	Maps a host name such as mail.adatum.com to an IP address, such as 131.107.10.10.
CNAME	Canonical name	Points one host record, such as adatum.ftp.adatum.com, to another host record, such as mail.lucernepublishing.com, or even another host record in another domain, such as www.contoso.com.
MX	Mail exchange	Points to the host that will receive mail for that domain. MX records must point to an A record, not to a CNAME record.
NS	Name server	Delegates a DNS zone to the specified authoritative name server.
SOA	Start of Authority	Defines the authoritative record for the zone.
SRV	Service	Locates hosts that are providing specific services, such as the Session Initiation Protocol (SIP) endpoint.
TXT	Text	Records a human-readable text field in DNS.

To Test the name resolution

- **Ipconfig** /all
- **ping** <host name>
- **nslookup** <host name> <name server name>
- **nslookup** www.bestazuretraining.com ns1-01.azure-dns.com

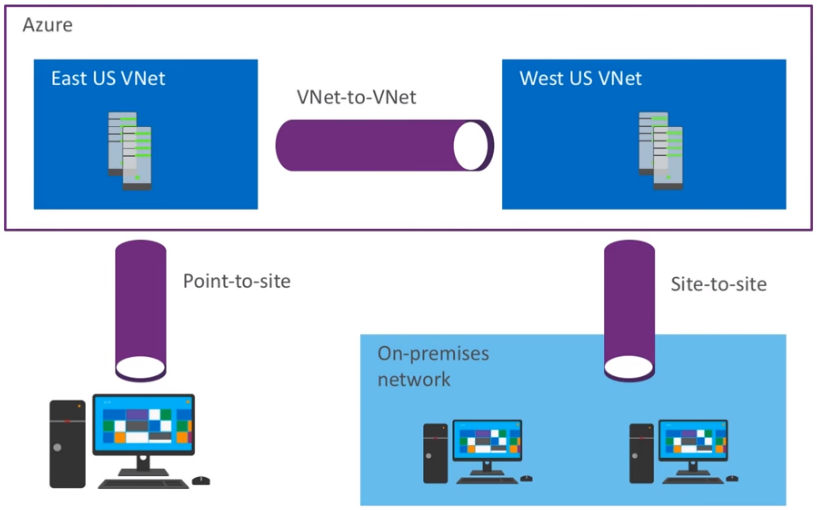
Private DNS Zones

https://docs.microsoft.com/en-us/azure/dns/private-dns-getstarted-cli

- A Private DNS Zone can be connected to multiple Virtual Networks (in any ADTenant/Subscriptions/Region).
- Private DNS Zone is a Global Services (Independent of Region)
- Here Domain Names are mapped **Private** IP Address.

Create connectivity between virtual networks

There are multiple ways to connect VNets. The sections below describe different ways to connect virtual networks.



Cloud-Only Virtual Networks:

You can choose not to make any kind of virtual private network (VPN) connection to a VNet. Instead, when you create a VM or cloud service, you can specify endpoints that external clients can connect to. An endpoint is a VIP and a port number. Therefore an endpoint can be used only for a specific protocol, such as connecting a Remote Desktop Protocol (RDP) client or browsing a website. These VNets are known as cloud-only virtual networks. A dynamic routing gateway is not required in the VNet. Endpoints are published to the Internet, so they can be used by anyone with an Internet connection, including your on-premises computers.

Point-to-Site VPNs

A simple way to connect a VPN to an Azure VNet is to use a Point-to-Site VPN. In these VPNs, you configure the connection on individual on-premises computers. No extra hardware is required but you must complete the configuration procedure on every computer that you want to connect to the VNet. Point-to-site VPNs can be used by the client computer to connect to a VNet from any location with an Internet connection. Once the VPN is connected, the client computer can access all VMs and cloud services in the VNet as if they were running on the local network.

VNet-to-VNet

Connecting a virtual network to another virtual network (VNet-to-VNet) is similar to connecting a virtual network to an on-premises site location. Both connectivity types use a VPN gateway to provide a secure tunnel using IPsec/IKE. The VNets you connect can be in **different subscriptions** and **different regions**.

The difference between the S2S AND V2V connection types is the way the local network gateway is configured. When you create a VNet-to-VNet connection, you do not see the local network gateway address space. It is automatically created and populated. If you update the address space for one VNet, the other VNet automatically knows to route to the updated address space. Creating a VNet-to-VNet connection is typically faster and easier than creating a Site-to-Site connection between VNets.

You can combine VNet to VNet communication with multi-site configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.

VNet peering

You may want to consider connecting your VNets using VNet Peering. VNet peering does not use a VPN gateway and has different constraints. Additionally, VNet peering pricing is calculated differently than VNet-to-VNet VPN Gateway pricing.

Site-to-Site VPNs

To connect **all the computers** in a physical site to an Azure VNet, you can create a Site-to-Site VPN. In this configuration, you do not need to configure individual computers to connect to the VNet, **instead you configure a VPN device**, which acts as a gateway to the VNet.

When you use the Site-to-Site IPsec steps, you create and configure the local network gateways manually. The local network gateway for each VNet treats the other VNet as a local site. This lets you specify additional address space for the local network gateway in order to route traffic. If the address space for a VNet changes, you need to update the corresponding local network gateway to reflect that. It does not automatically update.

ExpressRoute

ExpressRoute is a service that enables Azure customers to create a dedicated connection to Azure, which does not connect through the public Internet. This contrasts with VPNs, which use encryption to tunnel securely through the public Internet. Because ExpressRoute connections are dedicated, they can offer faster speeds, higher security, lower latencies, and higher reliability than VPNs.

Create and Configure VNet Peering

Virtual network peering enables you to seamlessly connect two Azure virtual networks. Once peered, the virtual networks appear as one, for connectivity purposes.

The traffic between virtual machines in the peered virtual networks is routed through the **Microsoft backbone infrastructure**, much like traffic is routed between virtual machines in the same virtual network, through *private* IP addresses only.

Azure supports:

- **VNet peering** - connecting VNets within the **same Azure region.**
- **Global VNet peering** - connecting VNets across different Azure regions.

Benefits

1. Its best alternative to VPN for vNets because all network traffic between peered virtual networks is **private and routed over Azure internal networks** instead of public internet.

2. A low-latency, high-bandwidth connection between resources in different virtual networks.

3. The ability to transfer data across Azure subscriptions, deployment models, and across Azure regions.

Pros and Cons over VPN Gateway

Pros

1. Faster and easier to setup than VPN
2. No Public IP required.

Cons

1. Peering relationships are not transitive.

   If you create peerings between:

   - VirtualNetwork1 & VirtualNetwork2
   - VirtualNetwork2 & VirtualNetwork3

   There is no peering between VirtualNetwork1 and VirtualNetwork3 through VirtualNetwork2.

2. Cannot use overlapping address spaces.

Pricing:

https://azure.microsoft.com/en-us/pricing/details/virtual-network/

Configuring a Peering

1. Select the Vnet → Settings → **Peerings**

- This virtual network, Peering link name = East-to-West
- Remote virtual network, Peering link name = West-to-East
- Virtual network = Demo-westus-rg
- Add

2. Verify that Peering is showing status=**Connected**